Position Paper: Supporting Novel Accelerators

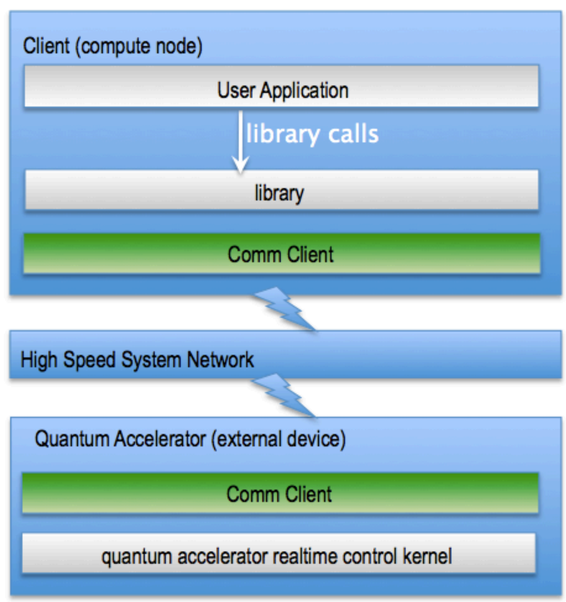
Terry Jones[1], Travis Humble[1], Larry Kaplan[2]

{trj, humblets} @ ornl.gov[1]   lkaplan @ cray.com[2]

Problem statement:
Despite the advances of traditional binary-based computers and the algorithms than run on them, quick solutions to many problems remain elusive. Quantum computing researchers have told us for years that, for certain problems, quantum computers can solve in seconds what would take a binary computer years to solve. In the past, however, quantum computers faced a significant problem: actual hardware was too difficult to construct.

Quantum computers are now a reality. Moreover, the initial customers are proving to be similar to the traditional DOE/ASCR user community -- the Canadian company D-Wave Systems recently sold a large quantum computer to Lockheed Martin. While current incarnations have a very narrow scope of suitable problems (problems which are both interesting and applicable to their special capabilities), future generations should enlarge both the number of applicable problems as well as the difficulty of the problem. This proposal seeks to establish the necessary system software infrastructure to support accelerators based on quantum computer concepts. Such accelerators would dramatically improve time-to-solution for known classes of important problems such as finding minima of complex systems (useful for applications like protein folding or certain graph problems) and other NP-Hard problems.

Proposed approach:
We propose establishing the infrastructure to support novel accelerators including quantum computing accelerators. The optimal way to interface scientific applications with such accelerators from a parallel application is a research question we would like to investigate. Initially, we imagine focusing on two approaches. First, we would investigate a library-based approach as a viable option (see figure at right). Second, we would also investigate providing support for programming model integration, possibly in the form of OpenACC or OpenMP extensions.



Some of the specific areas of runtime investigation include: data representation, and conversion and movement between domains, control, status management, error handling, program instantiation within the accelerator.

As an early step to the acceptance of quantum computers, we envision such accelerators augmenting traditional binary supercomputers in much the same way as vector accelerators appeared alongside the more sophisticated vector-based supercomputers of the mid-1970s. ORNL is developing a team of quantum computing specialists. Current ORNL advances have made the simulation of the quantum machine possible with cycle-accurate data for the controlling realtime computer. Other research is investigating the use of quantum computing for such varied tasks as protein folding and linear algebra problems.

To realize these initial steps for quantum computing, both hardware and software research is needed (including important system software aspects).

Related Work:
Although a staple problem in theoretical computer science for nearly two decades, the development of practical quantum computing architectures and system engineering methods remains largely unexplored [1]. Significant research has focused on the quantum computational hardware, i.e., the qubits and gates underlying computations as well as the low-level protocols responsible for fault-tolerant computation based on quantum error correction techniques [2].

However, the system level design of even these fundamental components remains unknown [3] and much more so for their integration with traditional supercomputing systems. Some work has been devoted to the design of quantum programming languages and quantum circuit design tools [4]. But these descriptions are either entirely independent of the hardware (languages) or entirely specific to a given hardware (circuit layout), and the abstraction of quantum computation into traditional supercomputing environments remains at large.

Assessment:
While only applicable to certain problems, quantum computing provides a homerun for those problems. And those problems are important. The full benefit of providing a usable interface to quantum computing is unknown, but certainly is ripe with potential.

At the algorithmic level, quantum computing is useful for solving linear systems of equations, finding molecular properties (including ground state energies and thermal rate constants), simulating quantum field theories, and solving outstanding problems in graph theory. Successful implementations, however, will be decided by whether these algorithms can be adapted to meaningful problem sizes and integrated alongside existing or future computing platform (as co-processors/accelerators). This includes estimates of the computational complexity (as opposed to algorithmic complexity), which can only be accomplished with respect to a given architecture.

Anticipating the emergence of quantum-enabled processors within the next 10 years motivates the need to evaluate what additional requirements quantum processors place on future operating systems and runtime software. Quantum-enabled processors are necessarily unique from their classical counterparts, as they require real-time feedback-driven control of hardware and enforce causal constraints on scheduling and routing, such as the no-cloning principle (i.e., no fan-out). These subtle differences may sufficiently differentiate quantum processors as to demand additional requirements on runtime capabilities.

Despite initial interests in quantum computing for cryptographic purposes, little effort has been afforded to the evaluation of how quantum computing would impact the broad DOE mission space. Quantum algorithmic improvements in solving linear systems suggest a significant impact on the DOE simulation and modeling mission, while molecular properties and graph algorithms support material characterization and data analytics, respectively. Like previous technologies that have multiple potential uses (lasers, fiber optics, supercomputers, …), we feel that the general capabilities of these tools should go forward to impact the full array of scientific inquiry supported by DOE/ASCR through programs like INCITE and ALCC.

Quantum computing offers much in terms of uniqueness and applicability. It has the property that it can do reversible computation (once you observe, its no longer reversible – but this is the only place it is irreversible). A reasonable interface to reversible computing may offer many interesting possibilities from starting from a desired state and working backward in the search for suitable initial conditions, to advanced debugging capabilities.

References:

1. K.M. Svore; A.V. Aho; A.W. Cross; I. Chuang; I.L. Markov; , "A layered software architecture for quantum computing design tools," *Computer* , vol.39, no.1, pp.74-83, Jan. 2006 doi: 10.1109/MC.2006.4; http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1580386&isnumber=33375

2. T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, J. L. O/'Brien, “Quantum computers”, Nature, **464**, 45-53 (4 March 2010). doi:10.1038/nature08812 http://www.nature.com/nature/journal/v464/n7285/abs/nature08812.html

3. N. C. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, Y. Yamamoto, “Layered architecture for quantum computing”, arXiv:1010.5022v2, http://arxiv.org/abs/1010.5022

4. M. Whitney, N. Isailovic, Y. Patel, and J. Kubiatowicz. 2007. Automated generation of layout and control for quantum circuits. In *Proceedings of the 4th international conference on Computing frontiers* (CF '07). ACM, New York, NY, USA, 83-94. DOI=10.1145/1242531.1242546 http://doi.acm.org/10.1145/1242531.1242546